

Instructions to candidates

Number of Pages: Four (04)

Number of questions: One (01) practical question

Time allocation: One (01) hour and thirty (30) minutes

Mark allocation: Hundred (100) marks

You need to upload the project folder as a ZIP file to the given link on the VLE.

When uploading, rename the ZIP file with your 'Registration Number' as shown in the example (i.e., UWU_CST_20_XXX).

You can download the "resources" ZIP file from the VLE for the database queries and other resources.

1. Imagine you are the owner of an online electronics store that sells a wide range of electronic gadgets and accessories. You want to enhance the user experience on your website and streamline the process of ordering products for your customers. To achieve this, you decide to introduce a new and user-friendly "Product Order Form". The form will include the following fields:

- **Product Name:** A text input field where customers can enter the name of the electronic product they wish to purchase. For example, "Smartphone X" or "Wireless Earbuds."
- **Quantity:** A numeric input field where customers can specify the quantity of the product they want to buy. This helps you better manage stock and shipping requirements.
- **Shipping Address:** A textarea field where customers can provide their shipping address for smooth and accurate delivery. You emphasize the importance of providing a complete and correct address.
- **Preferred Delivery Days:** A set of checkboxes displaying the available delivery days - Monday, Wednesday, and Friday. Customers can check one or more days based on their preferences for receiving the product.
- **Payment Method:** A set of radio buttons with options for Credit Card, PayPal, and Bank Transfer. Customers can select their preferred payment method.
- **Submit Button:** A button labelled "Place Order." This will submit the form and initiate the order process.

Your web application's user interface looks like Figure 1. The basic template is provided in the resources ZIP file for your convenience. However, please note that this template is not fully complete. You will need to make the necessary modifications while developing the product order process.

Product Order Form

Product Name:

Quantity:

Shipping Address:

Preferred Delivery Days:

Monday
 Wednesday
 Friday

Payment Method:

Credit Card
 PayPal
 Bank Transfer

[Place Order](#)

Figure 1: Order Form User Interface

This implementation must follow Object Oriented Programming (OOP) concepts, which require the development of two classes: Order and DbConnector. The Order class serves as a representation of an order for an electronic product in an online electronics store. It encapsulates various properties and methods to effectively manage order information and interact with a database. The following is the description of the Order class:

- **Properties:** order_id (private integer), product_name (private string), quantity (private integer), shipping_address (private string), delivery_days (private array of strings), and payment_method (private string).
- **Constructor:** The constructor takes five parameters: product_name (string), quantity (integer), shipping_address (string), delivery_days (array of strings), and payment_method (string). It initializes the object's properties with the provided values.
- **Methods:** This class has only one method named "save". This method is responsible for saving the order information to a database. The SQL command to create the relevant database table is there in the "resources" zip file. It takes a database connection object as a parameter (db_connection) to interact with the database. The method performs the

necessary database operations to insert the order details into the appropriate table. It returns a boolean value indicating whether the order was successfully saved or not (true if successful, false otherwise).

The DbConnector class represents a database connection manager that facilitates connecting to a database using PDO (PHP Data Objects). It encapsulates the necessary properties and a method to establish a database connection. The following is the description of the DbConnector class:

- **Properties:** host (private string), dbname (private string), dbuser (private string), and dbpw (private string)
- **Methods:** This class has only one method named "getConnection". This method is responsible for creating and returning a PDO (PHP Data Objects) database connection. It establishes a connection to the specified database. The method returns a PDO object.

You are required to:

- a. Create a NetBeans PHP web application project.
(05 marks)
- b. Create an "index.php" PHP page under the project to display the Figure 1 interface, utilizing the provided template files.
(12 marks)
- c. Create a database named "my_shop" and execute the given SQL query from the 'SQL.txt' file in the "resources" ZIP file to create an "order" table.
(03 marks)
- d. Implement the following two (02) classes to handle the application's business logic, based on the provided class descriptions:
 - i. Order class – This class will handle the logic for saving new orders.
(40 marks)
 - ii. DbConnector class – This class will be used to connect the application with the database, facilitating the saving logic of a new order.
(15 marks)
- e. Create a separate PHP web file named "process_order" to handle the submission of the Figure 1 form.
(18 marks)
- f. After successfully saving an order, redirect the user back to the index.php (Figure 01 interface) and display a green colour message such as "Your order has been saved successfully."
(05 marks)
- g. In case any errors occur during the order processing, redirect the user to the index.php page and display a red colour message stating, "An error occurred. Please try again.
(02 marks)

Note:

- It is crucial to implement proper error handling and data validation within your implementation to ensure the integrity and security of the database operations.
- Use PHP's built-in functions like implode(\$separator, \$array) or serialize(\$array) when saving arrays to the database.
- Additionally, it is essential to address any potential errors that may arise during the database connection process.

[End of Paper]